Sound Segmentation

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Project Report

Submitted by

Dhruv Mittal (17BCE2110)
Achyut Tripathi (17BCE0954)

Slot: - C1

Subject: - Parallel and Distributed Computing

Documentation. Under the Guidance of: -
Prof. Saira Banu J

Abstract

Our project deals with modifying and building an already existing audio clip most preferably a song & improving the over all sound quality to the next level giving effects like clarity & background sound. We are using Keras package of python to segment the audio file. We aim to distinguish the heart beats from the noise present into the audio file which can help physicians to study the noise free heart sound.

We are using multiprocessing package of python to parallelize the looping statements of the project for speedup purposes.

Introduction

Theoretical Background

Sound engineering is more than just fiddling with knobs on a fancy-looking soundboard.

From microphone technicians to audio production managers, the goal of sound engineers is to use their technical skills and knowledge to produce crystal clear sound for any recording or broadcast program, on a variety of platforms.

The job leads you to work with a wide variety of production professionals including musical artists, movie directors and editors, television/news crew, studio managers and radio presenters.

Aim of proposed work

The project aims at showing the various steps involved in sound formation and later on sound enhancing. We aim to distinguish the heart beats from the noise present into the audio file which can help physicians to study the noise free heart sound.

Objectives of proposed work

In accordance with our project aim, we had to choose a sound engineering tool that satisfies the following criteria: -

- Easy to implement
- Easy to use
- Fast processing of the graphs (sound waves) (Time complexity should be less)
- System requirements should be minimal
- Less space and memory utilization

The tool chosen for merging multiple sounds as one is: multiprocessing package and keras.

Literature Survey

Survey of Models

We went through a lot of papers that treated sound as waves and papers which treated big data and ran it on distributed systems. There was one problem though in most of them. None of the data papers had anything to do with sound as a data-type and none of the sound engineering papers didn't perceive sound as a data type to be introduced in a sound enhancing environment most of the modification/enhancing was done either manually or with environments that involved mechanical intervention(medium change/dispersion). Hence we decided to play with the mathematical properties of sound and use the recursive graph as a dataset and their input characteristics as our data set. This data set will be massive and hence we will have to apply big data mining using python and then make a distributed sound environment and then play the music.

Audio Classification

Classification is probably the most important problem in machine learning applications. It refers to the task of classifying an unknown sample (in our case audio signal) to a set of predefined classes, according to some trained supervised model. The library provides functionalities for the training of supervised models that classify either segments or whole audio recordings. Support vector machines and the k-Nearest Neighbour classifier have been adopted towards this end. In addition, a cross-validation procedure is provided in order to extract the classifier with optimized parameters. In particular, the precision and recall rates, along with the F1 measure are extracted per audio class. Parameter selection is performed based on the best average F1 measure. High-level wrapper functions are provided so that the

feature extraction process is also embedded in the classification procedure. In this way, the users can directly classify unknown audio files or even groups of audio files stored in particular paths.

Audio Regression

Regression is the task of estimating the value of an unknown variable (instead of distinct class labels), given a respective feature vector. It can also be rather important in the context of an audio analysis application, in cases there are mappings from audio features to a real-valued variable. A typical example is speech emotion estimation, where the emotions are not represented by discrete classes (e.g. Anger, Happiness, etc) but by dimensional representations (e.g. Valence—Arousal). The library supports SVM regression training in order to map audio features to one or more supervised variables. In order to train an audio regression model the user should provide a series of audio segments stored in separate files in the same folder. In addition, the user must provide a comma-separated-file (CSV), where the respective ground truth values of the output variable are stored. During the training phase, for each CSV file a separate variable is trained. Note that the regression training functionality also contains a parameter tuning procedure, where a cross-validation evaluation is performed, similar to that of the classification case. However, the performance measure maximized in the context of the regression training procedure is the Mean Square Error (MSE). In addition to that, for each tested parameter value, the MSE of the training data is also calculated to provide a measure of "overfitting". Finally, the parameter tuning procedure returns the MSE of the "average estimator", i.e. a method that always returns the average value of the estimated parameter (based on the training data), in order to provide a baseline performance measure. This is equivalent to the "random classifier" used as a worst-case performance when evaluating classification methods.

Audio Segmentation

Audio segmentation focuses on splitting an uninterrupted audio signal into segments of homogeneous content. The term "homogeneous" can be defined in many different ways, therefore there exists an inherent difficulty in providing a global definition for the concept. The library provides algorithmic solutions for two general subcategories of audio segmentation:

- the first contains algorithms that adopt some type of "prior" knowledge, e.g. a pre-trained classification scheme. For that type of segmentation the library provides a fix-sized joint segmentation—classification approach and an HMM-based method.
- the second type of segmentation is either unsupervised or semi-supervised. In both cases, no prior knowledge on the involved classes of audio content is used. Typical examples are silence removal, speaker diarization and audio thumbnailing.

Supervised audio segmentation

Fix-sized segmentation

This straightforward way of segmenting an audio recording to homogeneous segments splits the audio stream into fixed-size segments and classifies each segment separately using some supervised model. Successive segments that share a common class label are merged in a post processing stage. In addition, the library extracts some basic statistics.

Table of Literature Review

Author	About	Merits and Demerits	Year of Publication
Amandine Pras **From sound production to music engineering Paper**	This paper reflects upon the studio educational needs of musicians who want to learn how to record their own projects. It builds on a mixed-method investigation of studio professional's contributions to musical recordings in the digital era, which is synthesized in a chapter of Music, Technology & Education: Critical Perspectives edited by King and Himonides. We will extend the outcomes of this investigation with recent audio examples and a case study involving young musician-engineers in New York who use audio technology in symbiosis with their music creation. Eventually, a claim will be made regarding the necessity of teaching in music programs the listening and artistic skills required to work in the studio.	Explains the difference between mastering and music making as a whole and gives a description of good practices for making music and on how to compare two audios.	2016
Samantha Bennett **Special Issue on Augmented and Participatory Sound and Music Interaction using Semantic Audio**	Professional audio recording practices and multimedia methods on expansion, editing and mastering.	Gives an Idea about multimedia operations	2018

Laurent Daudet **Audio sparse decompositions in parallel**	Slight modification of MP makes it possible to break down the decomposition into multiple local tasks, while avoiding blocking effects. His simulations on audio signals indicate that this Parallel Local Matching Pursuit (PLoMP) gives results comparable to the original MP algorithm, but could potentially run in a fraction of the time — onthe-fly sparse approximations of highdimensional signals should soon become a reality.	Applies MP modification on signals and not the dataset we looked for.	2015
Prof. Stewart W s **Parallel orithm Design**	In this model, a parallel program is viewed as a collection of tasks that communicate by sending messages to each other through channels. A task consists of an executable unit (think of it as a program), together with its local memory and a collection of I/O ports. The local memory contains program code and private data, i.e., the data to which it has exclusive access. An access to this memory is called a local data access. The only way that a task can send copies of its local data to other tasks is through its output ports, and conversely, it can only receive data from other tasks through its input ports. An I/O port is an abstraction; it corresponds to some memory location that the task will use for sending or receiving data. Data sent or received through a channel is called a nonlocal data access	Defines well how tasks should be executed a as whole but fails to be specific	2016

Marc Aurel Kiefer, Korbinian Molitorisz, Jochen Bieler, Walter F. Tichy **Parallelizing a Real-time Audio Application - A Case Study in Multithreaded Software Engineering**	Case study on parallelizing commodity software with over 700,000 lines of code. In contrast to best practice guidelines, they investigate what parallelization strategy can effectively be used in data stream-intensive applications. Performing an in-depth analysis of the software architecture and its run-time performance, we locate parallelization potential and propose three different parallelization strategies. They evaluate them with respect to their parallel performance impact. Regarding the application'sintrinsic realtime requirement and a very short stream cycle turnaround time, a busywaiting strategy offers the best	They used multicore parallelisation for software based methods on big data but did not manage sound as a whole.	2014
	performance of 327 µs per cycle on an eightcore machine. With an efficiency of 99% this is close to the optimal schedule.		
Saadia Zahid **Optimized Audio Classification and Segmentation Algorithm by Using Ensemble Methods**	Audio segmentation is a basis for multimedia content analysis which is the most important and widely used application nowadays. An optimized audio classification and segmentation algorithm is presented in this paper that segments a superimposed audio stream on the basis of its content into four main audio types: pure-speech, music, environment sound, and silence. An algorithm is proposed that preserves important audio content and reduces the misclassification rate without using large amount of training data, which handles noise and is suitable for use for real-time applications. Noise in an audio stream is segmented out as environment sound.	Use of ANN which very hard to implement even for simple project.	2015

	Two audio classification systems have been proposed in this work in which an audio stream is discriminated into homogenous regions and classified into basic audio types such as speech, non-speech, silence, music, environmental sounds and so on.While, in the retaining levels of two audio classification systems, one of the algorithms KNN, SVM, and GASOM has been used as a classifier.	Use of KNN, an old algorithm.	2018
Lie Lu, Hong-Jiang Zhang, Senior Member, IEEE, and Hao Jiang **Content Analysis for Audio Classification and Segmentation**	Audio classification is processed in two steps, which makes it suitable for different applications. The first step of the classification is speech and nonspeech discrimination. In this step, a novel algorithm based on K-nearest-neighbor (KNN) and linear spectral pairs-vector quantization (LSP-VQ) is developed. The second step further divides nonspeech class into music, environment sounds, and silence with a rule-based classification scheme.	Experimental evaluation has shown that the proposed audio classification scheme is very effective and the total accuracy rate is over 96%. The novel scheme and new features introduced ensure that the system can achieve high accuracy even with a smaller testing unit.	2002
Xu-Kui Yang, Liang He, Dan Qu, Wei-Qiang Zhang & Michael T. Johnson **Semi-supervised feature selection for audio classification based on constraint compensated Laplacian score**	In this paper, we present a semi-supervised feature selection algorithm named Constraint Compensated Laplacian score (CCLS), which takes advantage of the local geometrical structure of unlabeled data as well as constraint information from labeled data. We apply this method to the audio classification task and compare it with other known feature selection methods. Experimental results demonstrate that CCLS gives substantial improvement.	The performance of CCLS was not as good as that of ReliefF in terms of optimized number of features. This may indicate that there are some redundancy features in the optimum feature set selected by CCLS method.	2016

Dabbabi Karim, Cherif Adnen **An Optimization of Audio Classification and Segmentation using GASOM Algorithm**	In this paper, we present an optimized audio classification and segmentation algorithms that are used to segment a superimposed audio stream according to its content into 10 main audio types: speech, non-speech, silence, male speech, female speech, music, environmental sounds, and music genres, such as classic music, jazz, and electronic music. We have tested the KNN, SVM, and GASOM algorithms on two audio classification systems. In the first audio classification system, the audio stream is discriminated into speech no-speech, purespeech/silence, male speech/female speech, and music/ environmental sounds. However, in the second audio classification system, the audio stream is segmented into music/speech, pure-speech/silence, male speech/female speech.	Experimental results have shown that the GASOM algorithm is so efficient for most audio discrimination types in terms of accuracy and time consumption. Thus, this advantage plus the no-requirement of much training data makes this algorithm very useful for real-time multimedia applications.	2018
Shawn Hershey, Sourish Chaudhuri **CNN ARCHITECTURES FOR LARGE-SCALE AUDIO CLASSIFICATION**	Convolutional Neural Networks (CNNs) have proven very effective in image classification and show promise for audio. We use various CNN architectures to classify the soundtracks of a dataset of 70M training videos (5.24 million hours) with 30,871 video-level labels.	The state-of-the-art image networks are capable of excellent results on audio classification when compared to a simple fully connected network or earlier image classification architectures. In Section 4.2 we saw results showing that training on larger label set vocabularies can improve performance, albeit modestly, when evaluating on smaller label sets.	2017
Honglak Lee Yan Largman **Unsupervised feature learning for audio classification using convolutional deep belief networks**	In this paper, we apply convolutional deep belief networks to audio data and empirically evaluate them on various audio classification tasks. In the case of speech data, we show that the learned features correspond to phones/phonemes. In addition, our feature representations learned from unlabeled audio data show very good performance for multiple audio classification tasks.	Modern speech datasets are much larger than the TIMIT dataset. While the challenge of larger datasets often lies in considering harder tasks, our objective in using the TIMIT data was to restrict the amount of labeled data our algorithm had to learn from.	2010

Serkan Kiranyaz,Moncef Gabbouj **A generic audio classification and segmentation approach for multimedia indexing and retrieval**	We present a fuzzy approach toward hierarchic audio classification and global segmentation framework based on automatic audio analysis providing robust, bi-modal, efficient and parameter invariant classification over global audio segments. The input audio is split into segments, which are classified as speech, music, fuzzy or silent	We have achieved good results with respect to our primary goal of being able to minimize the critical errors on audio content classification by introducing fuzzy modeling in the feature domain and shown the important role of having the global and perceptually meaningful segmentation on the accurate classification (and vice versa) in this context. The proposed work achieves significant advantages and superior performance over existing approaches for automatic audio content analysis, especially, in the context of audio-based indexing and retrieval for large-scale multimedia databases.	2006
Catherine Guastavino **IMPROVING THE SOUND QUALITY OF RECORDINGS THROUGH COMMUNICATION BETWEEN MUSICIANS AND SOUND ENGINEERS**	Sound engineers were expected to make appropriate sound choices by taking into consideration the musicians' requests. However, the communication flow between musicians and sound engineers appears to be a critical issue. Therefore, we propose a method that aims to help musicians come to a consensus on their expectations regarding sound quality and communicate these expectations to the sound engineer.	The proposed method helps musicians come to a consensus within the band and make their sound quality wishes explicit to the sound engineers. This method also helps sound engineers define a specific sound objective and stimulates their creativity to achieve this goal.	2009

Qingshu Liu, Xiaomei Wu & Xiaojing Ma **An automatic segmentation method for heart sounds**	There are two major challenges in automated heart sound analysis: segmentation and classification. An efficient segmentation is capable of providing valuable diagnostic information of patients. In addition, it is crucial for some feature-extraction based classification methods. Therefore, the segmentation of heart sound is of significant value.	The proposed method shows reliable performance on the segmentation of heart sounds. Compared with previous works, this method can be applied to not only normal heart sounds, but also the sounds with S3, S4 and murmurs, thus greatly increasing the applied range.	2018

Implementation Details

Segmentation, specially for audio data analysis, is an important pre-processing step. This is because we can segment a noisy and lengthy audio signal into short homogeneous segments, which are handy short sequences of audio used for further processing. Now to solve a segmentation problem, we can either do it directly using unsupervised methods or convert it into a supervised problem and then group it according to its class.

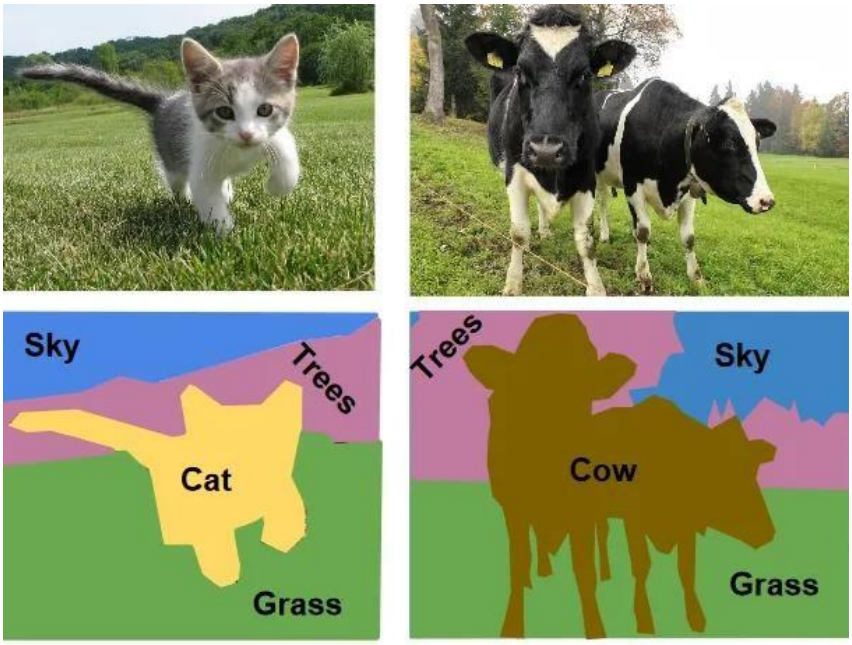
To explain this more intuitively, lets take an example of Image Segmentation task.

Suppose we have an image of a cat in a field as we can see below. What you want is to divide the image into chunks – so that one individual object can be separately identified from the other. We can do this in two ways

- **Approach 1:** From each pixel of the image, find out the pixels which are close to each other and have an approximately similar color. We can cluster these pixels together to form a bigger picture of an object. In the example below, our cat is mostly greyish white. So it would be easier to find the pixels and

segmenting the cat out of the image. This is an **unsupervised approach** to segmentation.

- **Approach 2:** Train a model by giving it explicit examples of the classes belonging to the image – specifically a cat, trees and sky. Then get the model predictions on which class is present where in the image. This is a **supervised approach** to segmentation.



We will be using Keras package from the python library to do the computation in the audio data.

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolutional networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU.

On the other hand, to process the audio signal into wavelets we are using librosa package from the python library. LibROSA is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.

The task in the challenge is to find a method that can locate sounds particular to a heart (aka lub & dub, which are technically called S1 and S2) within audio data and then segment the audio files on the basis of these sounds. After segmenting the sounds, the challenge then asks us to produce a method that can classify heartbeat into normal and diseased categories. For the purpose of this article, we will take up only the first task of the challenge, i.e. to segment heart audio.

Conclusion

The automatic segmentation algorithm is found to be effective to segment phonocardiogram signals into four parts. The algorithm has shown 93 percent success in 37recordings, which include 515 cycles heart sounds. This is a good basis for further analysis of the heart sound signals. With this segmentation, we can extract the features of each segment, such as the root mean square, the peak intensity, the peak location, the duration, the split the interval of S2, etc. Wecan also do other processes to each segment for the classification.

Reference

I] Sapire DW. Understanding and diagnosing pediatric heart disease: Heart sounds and murmurs. Norwalk, Connecticut Applcton & Langc 1992: 27-43.

[2] Lehner RJ, Rangayyan RM. A three-channel microcomputer system for segmentation and characterization of the phonocardiogram. IEEE Trans. on Biomedical Engineering 1987; 34:

[3] Groch *MW,* Domnanovich *JR,* Erwin WD, A new heartsounds gating devices for medical imaging. IEEE Trans. On Biomedical Engineering 1992; 39: 307-310.

[4] Iwata A, Ishii N, Suzumura N. Algorithm for detecting the fist and the second heart sounds by spectral tracking. Med. & Biol. Eng. & Comput Jan 1980: 19-26

[5]SI Gerbarg DS, etc. Analysis of phonocardiogram by 3 digital computer. Circulation Research 1962; 11: 569-576.

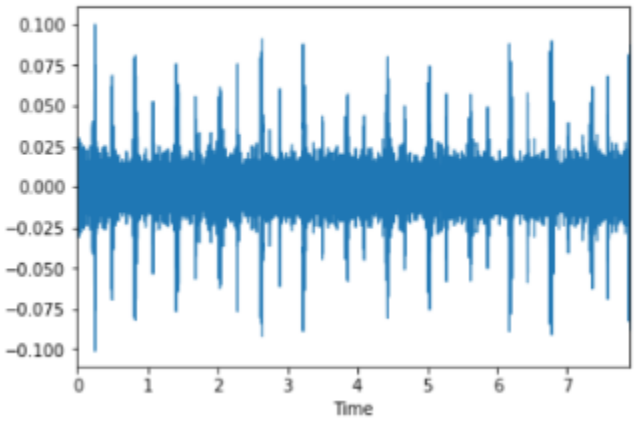
[6] Hurst *JW.* The heart arteries and veins, 7th ed. McGraw-Hill Information Services Company, New York 1990: Ch.

14: 175-242.

Output

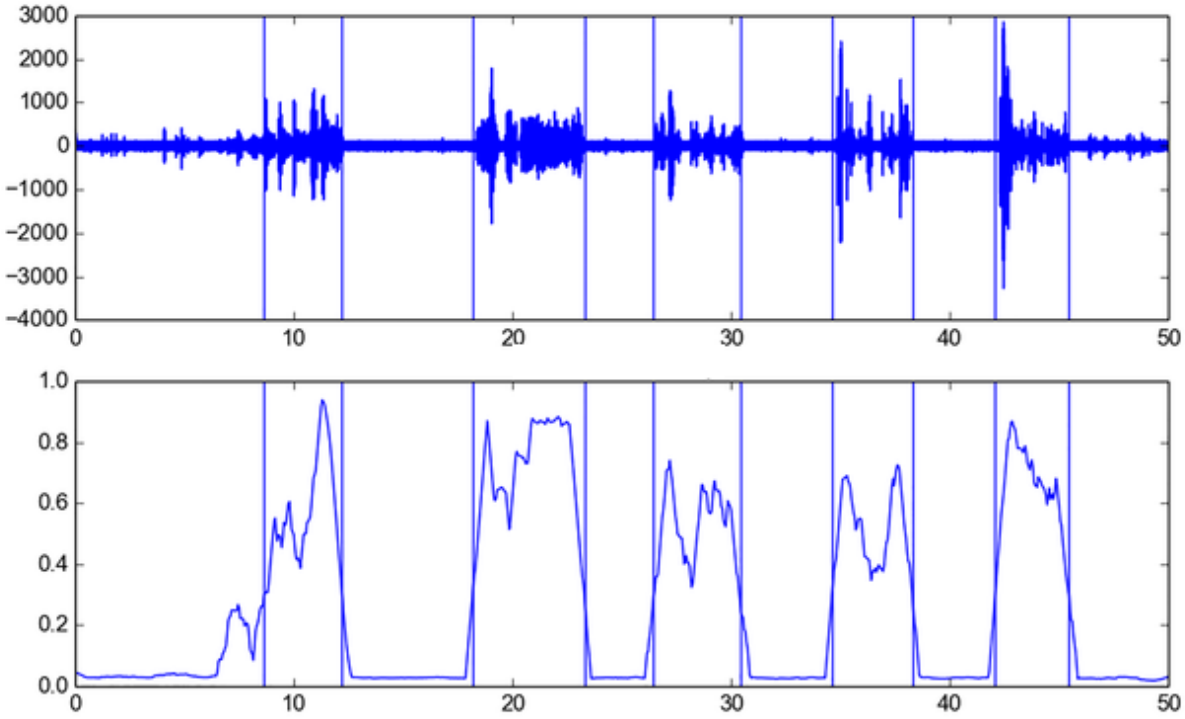
```
In [5]: display.waveplot(data, sr=sampling_rate)
```

```
Out[5]: <matplotlib.collections.PolyCollection at 0x1f78a74c278>
```




```
In [19]: model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 20000, 1)          0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 19991, 50)         550
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 2499, 50)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 2490, 50)          25050
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 312, 50)           0
_________________________________________________________________
flatten_1 (Flatten)          (None, 15600)             0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 15601
=================================================================
Total params: 41,201
Trainable params: 41,201
Non-trainable params: 0
_________________________________________________________________
```


3000
2000
1000
0
−1000
−2000
−3000
−4000
0
10
20
30
40
50
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50